



## **SISTEMA INTEGRADO DE GESTIÓN DE EXPEDIENTES MODULAR (SIGM)**

**INTERFAZ DE CONEXIÓN DE GESTION DE TERCEROS  
(REGISTRO PRESENCIAL)**

**SIGM v3**



**Administración Local Soluciones**

### Control de versiones

| Versión | Fecha edición | Cambio producido   | Autor  |
|---------|---------------|--|--------|
| 01      | 15-06-2009    | Versión inicial  | IECISA |
| 02      | 24-06-2009    | Eliminado el método getEAddress<br>Añadido el método getDocsType | IECISA |
| 03      | Octubre 2012  | Actualización para SIGM 3.0                                      | IECISA |

## ÍNDICE

|          |  |           |
|----------|--|-----------|
| <b>1</b> | <b>INTRODUCCIÓN .....</b>  | <b>4</b>  |
| 1.1      | VISIÓN GENERAL DEL SISTEMA.....                                  | 4         |
| 1.2      | FINALIDAD DEL DOCUMENTO .....                                    | 4         |
| 1.3      | DEFINICIONES Y ABREVIATURAS.....                                 | 4         |
| <b>2</b> | <b>INTERFAZ.....</b>   | <b>5</b>  |
| 2.1      | MÉTODO CREATE .....  | 6         |
| 2.2      | MÉTODO UPDATE .....  | 6         |
| 2.3      | MÉTODO SEARCH.....   | 6         |
| 2.4      | MÉTODO COUNT .....   | 6         |
| 2.5      | MÉTODO GETINFO.....  | 6         |
| 2.6      | MÉTODO GETADDRESSES .....  | 8         |
| 2.7      | MÉTODO GETADDRESSES .....  | 8         |
| 2.8      | MÉTODO GETPROVINCES.....   | 8         |
| 2.9      | MÉTODO GETCITIES .....   | 8         |
| 2.10     | MÉTODO GETDOCSTYPE.....  | 8         |
| 2.11     | MÉTODO GETADDRESSESTYPE.....                                     | 9         |
| <b>3</b> | <b>ANEXOS. ESTRUCTURAS PARÁMETROS DE ENTRADA Y SALIDA.....</b>   | <b>10</b> |
| 3.1      | ESTRUCTURA DE LA CADENA XML DE INFORMACIÓN DE PERSONAS .....     | 10        |
| 3.2      | ESTRUCTURA DE LA CADENA XML DE CRITERIOS DE BÚSQUEDA.....        | 14        |
| 3.3      | ESTRUCTURA DE LA CADENA XML DE IDENTIFICADOR DE OBJETO.....      | 16        |
| 3.4      | ESTRUCTURA DE LA CADENA XML DE LISTA DE PERSONAS .....           | 17        |
| 3.5      | ESTRUCTURA DE LA CADENA XML DE LISTA DE DOMICILIOS .....         | 19        |
| 3.6      | ESTRUCTURA DE LA CADENA XML DE LISTA DE PROVINCIAS .....         | 21        |
| 3.7      | ESTRUCTURA DE LA CADENA XML DE LISTA DE CIUDADES .....           | 22        |
| 3.8      | ESTRUCTURA DE LA CADENA XML DE LISTA DE TIPO DE DOCUMENTOS ..... | 23        |
| 3.9      | ESTRUCTURA DE LA CADENA XML DE LISTA DE TIPO DE DIRECCIÓN .....  | 24        |

## **1 Introducción**

### **1.1 Visión general del sistema**

AL SIGM es la plataforma de Tramitación Electrónica del MINETUR, solución integral para la tramitación electrónica de los procedimientos administrativos, que fomenta la interoperabilidad entre administraciones mediante su adaptación a estándares de comunicación así como la reutilización de recursos e información pública.

### **1.2 Finalidad del documento**

El objeto de este documento, es describir la implementación de la interfaz de conexión de gestión de Terceros del Registro de Entrada / Salida, del proyecto SIGEM "SISTEMA INTEGRADO DE GESTIÓN MUNICIPAL".

### **1.3 Definiciones y Abreviaturas**

A continuación se expone una tabla con los diferentes acrónimos y abreviaturas utilizados a lo largo del documento, con su correspondiente definición.

| <b>Acrónimo / Abreviatura</b> | <b>Definición</b>                                   |
|-------------------------------|---|
| MINETUR                       | Ministerio de Industria, Energía y Turismo          |
| IECISA                        | Informática El Corte Inglés S.A.                    |
| SIGM                          | Sistema Integrado de Gestión de Expedientes Modular |
| AL                            | Administración Local                                |

## 2 Interfaz

Package: ***com.ieci.tecdoc.common.adapter***

Nombre de interfaz: ***PersonValidation***

Métodos:

```
public String create (String xmlPersonInfo) throws  
Exception;  
  
public String update (String xmlPersonInfo) throws  
Exception;  
  
public String search (String xmlSearchParameters)  
throws Exception;  
  
public LONG count (String xmlSearchParameters) throws  
Exception;  
  
public String getInfo (String xmlParamId) throws  
Exception;  
  
public String getAddresses(String xmlParamId) throws  
Exception;  
  
public String getAddresses(String xmlParamId, int  
typeAddress) throws Exception;  
  
public String getProvinces(String xmlParamId) throws  
Exception;  
  
public String getCities (String xmlParamId) throws  
Exception;  
  
public String getDocType (String xmlParamId) throws  
Exception;
```

## 2.1 **Método create**

Uso : Dar de alta una nueva persona.

Parámetros de entrada :

**xmlPersonInfo**: XML con la información de la nueva persona (Ver anexo)

Parámetros de salida : Identificador de la persona creada (Formato String)

## 2.2 **Método update**

Uso : Modificar los datos de una persona.

Parámetros de entrada :

**xmlPersonInfo**: XML con toda la información de la persona, incluyendo los cambios efectuados (Ver anexo)

Parámetros de salida : Identificador de la persona modificada (Formato String)

## 2.3 **Método search**

Uso : Realizar una búsqueda de personas, en base a los criterios de búsqueda indicados.

Parámetros de entrada :

**xmlSearchParameters**: XML con los criterios de búsqueda (Ver anexo)

Parámetros de salida : XML con la información de las personas que cumplen el criterio de búsqueda (Ver anexo)

## 2.4 **Método count**

Uso : Realizar una búsqueda de personas, en base a los criterios de búsqueda indicados, y devuelve el número de personas que cumplen los criterios indicados.

Parámetros de entrada :

**xmlSearchParametersEx**: XML con los criterios de búsqueda (Ver anexo)

Parámetros de salida : Número de personas encontradas que cumplan el criterio de búsqueda.

## 2.5 **Método getInfo**

Uso : Obtener toda la información de una persona.

Parámetros de entrada :

**xmlParamId**: XML con el identificador de la persona (Ver anexo)

Parámetros de salida : XML con toda la información de la persona indicada (Ver anexo)



## 2.6 **Método *getAddresses***

Uso : Obtener la lista de domicilios de una persona.

Parámetros de entrada :

**xmlParamId**: XML con el identificador de la persona (Ver anexo)

Parámetros de salida : XML con la lista de domicilios de la persona indicada (Ver anexo)

## 2.7 **Método *getAddresses***

Uso : Obtener la lista de direcciones de una persona en función del tipo de dirección.

Parámetros de entrada :

**xmlParamId**: XML con el identificador de la persona (Ver anexo)

**typeAddress**: entero con el identificador del tipo de dirección (Ver anexo)

Parámetros de salida : XML con la lista de direcciones telemáticas de la persona indicada (Ver anexo)

## 2.8 **Método *getProvinces***

Uso : Obtener la lista de provincias.

Parámetros de entrada :

**xmlParamId**: XML con un identificador nulo (Ver anexo)

Parámetros de salida : XML con la lista de provincias (Ver anexo)

## 2.9 **Método *getCities***

Uso : Obtener la lista de ciudades de una provincia.

Parámetros de entrada :

**xmlParamId**: XML con el identificador de la provincia (Ver anexo)

Parámetros de salida : XML con la lista de ciudades de la provincia indicada (Ver anexo)

## 2.10 **Método *getDocsType***

Uso : Obtener la lista de tipos de documentos.

Parámetros de entrada :

**xmlParamId**: XML con el identificador del tipo de tercero (Ver anexo)

Parámetros de salida : XML con la lista tipos de documentos del tipo de tercero indicado (Ver anexo)



## 2.11 **Método *getAddressesType***

Uso : Obtener la lista de tipos de direccion.

Parámetros de entrada :

**xmlParamId**: XML con los datos obligatorios (Ver anexo)

Parámetros de salida : XML con la lista de tipos de documento (Ver anexo)

## 3 Anexos. Estructuras parámetros de entrada y salida

### 3.1 Estructura de la cadena XML de información de personas

Métodos: **create**, **update**, **getInfo**

DTD:

```
<!ELEMENT Persona (Id, Tipo, Nombre, Apellido1, Apellido2, TipoDoc, NIF, Domicilios?, Edirecciones?)>
```

```
<!-- Identificador de sesión -->
```

```
<!ATTLIST Persona sesionId CDATA #IMPLIED>
```

```
<!-- Indicador de si los datos pueden ser modificados -->
```

```
<!ATTLIST Persona bloqueado CDATA #IMPLIED>
```

```
<!-- Identificador de persona o de domicilio -->
```

```
<!-- No se especifica si la cadena XML se usa para dar de alta una persona -->
```

```
<!ELEMENT Id (#PCDATA)>
```

```
<!-- Tipo de persona -->
```

```
<!-- Valor 1 (Persona física), 2 (Persona jurídica) -->
```

```
<!ELEMENT Tipo (#PCDATA)>
```

```
<!-- Nombre de persona física o Razón social de persona jurídica -->
```

```
<!ELEMENT Nombre (#PCDATA)>
```

```
<!-- Primer apellido de persona física (no se informa para persona jurídica) -->
```

```
<!ELEMENT Apellido1 (#PCDATA)>
```

```
<!-- Segundo apellido de persona física (no se informa para persona jurídica) -->
```

```
<!ELEMENT Apellido2 (#PCDATA)>
```

```
<!-- Tipo de documento de identificación -->
```

```
<!ELEMENT TipoDoc (#PCDATA)>
```

```
<!-- Número de documento de identificación -->
```

```
<!-- Opcional si TypeDoc es nulo -->
```

```
<!ELEMENT NIF (#PCDATA)>
```

```
<!-- Lista de domicilios -->
```

```
<!-- Debe existir al menos un elemento en la lista -->
```

```
<!ELEMENT Domicilios (Domicilio+)>
```

```
<!-- Domicilio -->
```

```
<!ELEMENT Domicilio (Id, Direccion, Ciudad, CodPostal, Provincia, Preferencia)>
```

```
<!--Indica que el domicilio debe ser eliminado -->
```

```
<!ATTLIST Domicilio eliminar CDATA #IMPLIED>
```

```
<!--Dirección (sin normalizar) -->
```

```
<!ELEMENT Direccion (#PCDATA)>
```

```
<!-- Código de Población -->
```

```
<!ELEMENT Población (#PCDATA)>
```

```
<!-- Código postal -->
```

```
<!ELEMENT CodPostal (#PCDATA)>
```

```
<!-- Código de Provincia -->
```

```
<!ELEMENT Provincia (#PCDATA)>
```

```
<!-- Preferencia -->
<!-- Indica si el domicilio es el domicilio principal (Valor 1) -->
<!ELEMENT Preferencia (#PCDATA)>

<!-- Lista de direcciones telemáticas -->
<!-- Debe existir al menos un elemento en la lista -->
<!ELEMENT EDirecciones (Direccion+)>

<!--Dirección telemática -->
<!ELEMENT EDireccion (Id, Direccion, Tipo, Preferencia)>

<!--Indica que el domicilio debe ser eliminado -->
<!ATTLIST EDireccion eliminar CDATA #IMPLIED>

<!-- Identificador -->
<!ELEMENT Id (#PCDATA)>

<!--Dirección o número -->
<!ELEMENT Direccion (#PCDATA)>

<!--Tipo de dirección -->
<!ELEMENT Tipo (#PCDATA)>

<!-- Preferencia -->
<!-- Indica si la dirección es la dirección principal (Valor 1) -->
<!ELEMENT Preferencia (#PCDATA)>
```

## Esquema (XSD):

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema id="NewDataSet" xmlns="" xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns:msdata="urn:schemas-microsoft-com:xml-msdata">
  <xs:element name="Persona">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="Id" type="xs:string" minOccurs="0" />
        <xs:element name="Tipo" type="xs:string" minOccurs="0" />
        <xs:element name="Nombre" type="xs:string" minOccurs="0" />
        <xs:element name="Apellido1" type="xs:string" minOccurs="0" />
        <xs:element name="Apellido2" type="xs:string" minOccurs="0" />
        <xs:element name="TipoDoc" type="xs:string" minOccurs="0" />
        <xs:element name="NIF" type="xs:string" minOccurs="0" />
        <xs:element name="Domicilios" minOccurs="0" maxOccurs="unbounded">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="Domicilio" minOccurs="1" maxOccurs="unbounded">
                <xs:complexType>
                  <xs:sequence>
                    <xs:element name="Id" type="xs:string" minOccurs="0" />
                    <xs:element name="Direccion" type="xs:string" minOccurs="0" />
                    <xs:element name="Poblacion" type="xs:string" minOccurs="0" />
                    <xs:element name="CodPostal" type="xs:string" minOccurs="0" />
                    <xs:element name="Provincia" type="xs:string" minOccurs="0" />
                    <xs:element name="Preferencia" type="xs:string" minOccurs="0" />
                  </xs:sequence>
                </xs:complexType>
              <xs:attribute name="eliminar" type="xs:string" />
            </xs:sequence>
          </xs:complexType>
        </xs:element>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="EDirecciones" minOccurs="0" maxOccurs="unbounded">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="EDireccion" minOccurs="1" maxOccurs="unbounded">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="Id" type="xs:string" minOccurs="0" />
              <xs:element name="Direccion" type="xs:string" minOccurs="0" />
              <xs:element name="Tipo" type="xs:string" minOccurs="0" />
              <xs:element name="Preferencia" type="xs:string" minOccurs="0" />
            </xs:sequence>
          </xs:complexType>
          <xs:attribute name="eliminar" type="xs:string" />
        </xs:element>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:attribute name="sesionId" type="xs:string" />
  <xs:attribute name="bloqueado" type="xs:string" />
</xs:element>
<xs:element name="NewDataSet" msdata:IsDataSet="true" msdata:Locale="es-ES">
  <xs:complexType>
    <xs:choice maxOccurs="unbounded">
      <xs:element ref="Persona" />
    </xs:choice>
  </xs:complexType>
</xs:element>
</xs:schema>
```

### Ejemplo:

```
<?xml version="1.0" encoding="utf-8"?>
<Persona sesionId="65" bloqueada="0">
  <Id>12345</Id>
  <Tipo>1</Tipo>
  <Nombre>Manuel</Nombre>
  <Apellido1>Tavares</Apellido1>
  <Apellido2>Casanueva</Apellido2>
  <TipoDoc>2</TipoDoc>
  <NIF>45234567K</NIF>
  <Domicilios>
    <Domicilio>
      <Id>87657</Id>
      <Direccion>Calle Mayor 5 4º</Direccion>
      <Poblacion>23</Poblacion>
      <CodPostal>45678</CodPostal>
      <Provincia>1</Provincia>
      <Preferencia>1</Preferencia>
    </Domicilio>
  </Domicilios>
  <EDirecciones>
    <EDireccion>
      <Id>87657</Id>
      <Direccion>609765234</Direccion>
      <Tipo>1</Tipo>
      <Preferencia>1</Preferencia>
    </EDireccion>
    <EDireccion eliminar="1">
      <Id>87658</Id>
      <Direccion>mmm@uuu.com</Direccion>
      <Tipo>3</Tipo>
      <Preferencia>0</Preferencia>
    </EDireccion>
  </EDirecciones>
</Persona>
```

### 3.2 Estructura de la cadena XML de criterios de búsqueda

Métodos: **search**, **count**

DTD:

```
<!ELEMENT Criterios (Criterio*, ClausulaWhere?, ClausulaOrder?)>

<!-- Identificador de sesión -->
<!ATTLIST Criterios sesionId CDATA #IMPLIED>

<!-- Identificador de tipo de persona -->
<!-- Valor 1 (Persona física), 2 (Persona jurídica) -->
<!ATTLIST Criterios tipoPersona CDATA #IMPLIED>

<!--Índice de inicio en la lista de personas encontradas -->
<!ATTLIST Criterios inicio CDATA #IMPLIED>

<!--Rango de la lista de personas encontradas -->
<!ATTLIST Criterios rango CDATA #IMPLIED>

<!-- Domicilio -->
<!ELEMENT Criterio (Campo, Operador, Valor)>

<!-- Campo de búsqueda -->
<!ELEMENT Campo (#PCDATA)>

<!-- Operador de búsqueda -->
<!ELEMENT Operador (#PCDATA)>

<!-- Valor de búsqueda -->
<!ELEMENT Valor (#PCDATA)>

<!--Clausula Where adicional -->
<!ELEMENT ClausulaWhere (#PCDATA)>

<!--Clausula Order adicional -->
<!ELEMENT ClausulaOrder (#PCDATA)>
```

## Esquema (XSD):

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema id="NewDataSet" xmlns="" xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns:msdata="urn:schemas-microsoft-com:xml-msdata">
  <xs:element name="Criterios">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="ClausulaWhere" type="xs:string" minOccurs="0" msdata:Ordinal="1"
/>
        <xs:element name="ClausulaOrder" type="xs:string" minOccurs="0" msdata:Ordinal="2"
/>
        <xs:element name="Criterio" minOccurs="0" maxOccurs="unbounded">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="Campo" type="xs:string" minOccurs="0" />
              <xs:element name="Operador" type="xs:string" minOccurs="0" />
              <xs:element name="Valor" type="xs:string" minOccurs="0" />
            </xs:sequence>
          </xs:complexType>
        </xs:element>
      </xs:sequence>
      <xs:attribute name="sesionId" type="xs:string" />
      <xs:attribute name="tipoPersona" type="xs:string" />
      <xs:attribute name="inicio" type="xs:string" />
      <xs:attribute name="rango" type="xs:string" />
    </xs:complexType>
  </xs:element>
  <xs:element name="NewDataSet" msdata:IsDataSet="true" msdata:Locale="es-ES">
    <xs:complexType>
      <xs:choice maxOccurs="unbounded">
        <xs:element ref="Criterios" />
      </xs:choice>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

## Ejemplo:

```
<?xml version="1.0" encoding="utf-8"?>
<Criterios sesionId="65" personType="1" inicio="0" rango="10">
  <Criterio>
    <Campo>NIF</Campo>
    <Operador>Empieza por</Operador>
    <Valor>57829000j</Valor>
  </Criterio>
</Criterios>
```

### 3.3 Estructura de la cadena XML de identificador de objeto

Métodos: **getInfo**, **getAddresses**, **getProvinces**, **getCities**, **getDocsType**

DTD:

```
<!ELEMENT ParamId (Id)>

<!-- Identificador de sesión -->
<!ATTLIST ParamId sesionId CDATA #IMPLIED>

<!--Valor de identificador -->
<!ELEMENT Id (#PCDATA)>
```

Esquema (XSD):

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema id="NewDataSet" xmlns="" xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns:msdata="urn:schemas-microsoft-com:xml-msdata">
  <xs:element name="ParamId">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="Id" type="xs:string" minOccurs="0" msdata:Ordinal="0" />
      </xs:sequence>
      <xs:attribute name="sesionId" type="xs:string" />
    </xs:complexType>
  </xs:element>
  <xs:element name="NewDataSet" msdata:IsDataSet="true" msdata:Locale="es-ES">
    <xs:complexType>
      <xs:choice maxOccurs="unbounded">
        <xs:element ref="ParamId" />
      </xs:choice>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

Ejemplo:

```
<?xml version="1.0" encoding="utf-8"?>
<ParamId sesionId="65">
  <Id>2345</Id>
</ParamId>
```



### 3.4 Estructura de la cadena XML de lista de personas

Métodos: **search**

DTD:

```
<!-- Lista de personas -->
<!-- Debe existir al menos una persona en la lista -->
<!ELEMENT Personas (Persona?)>

<!--Índice de inicio en la lista de personas encontradas -->
<!ATTLIST Personas inicio CDATA #IMPLIED>

<!--Índice de fin en la lista de personas encontradas -->
<!ATTLIST Personas fin CDATA #IMPLIED>

<!-- Número total de personas encontradas -->
<!ATTLIST Personas total CDATA #IMPLIED>

<!-- Rango de personas devuelto -->
<!ATTLIST Personas rango CDATA #IMPLIED>

<!-- Persona -->
<!ELEMENT Persona (Id, Tipo, Nombre, Apellido1, Apellido2, TipoDoc, NIF)>

<!-- Identificador de persona o de domicilio -->
<!-- No se especifica si la cadena XML se usa para dar de alta una persona -->
<!ELEMENT Id (#PCDATA)>

<!-- Tipo de persona -->
<!-- Valor 1 (Persona física), 2 (Persona jurídica) -->
<!ELEMENT Tipo (#PCDATA)>

<!-- Nombre de persona física o Razón social de persona jurídica -->
<!ELEMENT Nombre (#PCDATA)>

<!-- Primer apellido de persona física (no se informa para persona jurídica) -->
<!ELEMENT Apellido1 (#PCDATA)>

<!-- Segundo apellido de persona física (no se informa para persona jurídica) -->
<!ELEMENT Apellido2 (#PCDATA)>

<!-- Tipo de documento de identificación -->
<!-- Persona física: N (NIF), P (Pasaporte), E (NIE), X (Otros), nulo -->
<!-- Persona jurídica: C (CIF), X (Otros), nulo -->
<!ELEMENT TipoDoc (#PCDATA)>

<!-- Número de documento de identificación -->
<!-- Opcional si TypeDoc es nulo -->
<!ELEMENT NIF (#PCDATA)>
```

## Esquema (XSD):

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema id="Personas" xmlns="" xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns:msdata="urn:schemas-microsoft-com:xml-msdata">
  <xs:element name="Personas" msdata:IsDataSet="true" msdata:Locale="es-ES">
    <xs:complexType>
      <xs:choice maxOccurs="unbounded">
        <xs:element name="Persona">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="Id" type="xs:string" minOccurs="0" />
              <xs:element name="Tipo" type="xs:string" minOccurs="0" />
              <xs:element name="Nombre" type="xs:string" minOccurs="0" />
              <xs:element name="Apellido1" type="xs:string" minOccurs="0" />
              <xs:element name="Apellido2" type="xs:string" minOccurs="0" />
              <xs:element name="TipoDoc" type="xs:string" minOccurs="0" />
              <xs:element name="NIF" type="xs:string" minOccurs="0" />
            </xs:sequence>
            <xs:attribute name="inicio" type="xs:string" />
            <xs:attribute name="fin" type="xs:string" />
            <xs:attribute name="total" type="xs:string" />
            <xs:attribute name="rango" type="xs:string" />
          </xs:complexType>
        </xs:element>
      </xs:choice>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

## Ejemplo:

```
<?xml version="1.0" encoding="utf-8"?>
<Personas inicio="0" fin="2" total="2" rango="20">
  <Persona>
    <Id>12345</Id>
    <Tipo>1</Tipo>
    <Nombre>Manuel</Nombre>
    <Apellido1>Manuel</Apellido1>
    <Apellido2>Manuel</Apellido2>
    <TipoDoc>N</TipoDoc>
    <NIF>45234567K</NIF>
  </Persona>
  <Persona>
    <Id>12645</Id>
    <Tipo>1</Tipo>
    <Nombre>José</Nombre>
    <Apellido1>López</Apellido1>
    <Apellido2>García</Apellido2>
    <TipoDoc>N</TipoDoc>
    <NIF>55214867J</NIF>
  </Persona>
</Personas>
```

### 3.5 Estructura de la cadena XML de lista de domicilios

Métodos: **getAddresses**

DTD:

```
<!-- Lista de domicilios -->
<!-- Debe existir al menos un elemento en la lista -->
<!ELEMENT Domicilios (Domicilio+)>

<!-- Domicilio -->
<!ELEMENT Domicilio (Id, Direccion, Población, CodPostal, Provincia, Preferencia)>

<!-- Identificador -->
<!ELEMENT Id (#PCDATA)>

<!-- Direccion -->
<!ELEMENT Direccion (#PCDATA)>

<!-- Código de Población -->
<!ELEMENT Población (#PCDATA)>

<!-- Código postal -->
<!ELEMENT CodPostal (#PCDATA)>

<!-- Código de Provincia -->
<!ELEMENT Provincia (#PCDATA)>

<!-- Preferencia -->
<!-- Indica si el domicilio es el domicilio principal (Valor 1) -->
<!ELEMENT Preferencia (#PCDATA)>
```

Esquema (XSD):

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema id="Domicilios" xmlns="" xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns:msdata="urn:schemas-microsoft-com:xml-msdata">
  <xs:element name="Domicilios" msdata:IsDataSet="true" msdata:Locale="es-ES">
    <xs:complexType>
      <xs:choice maxOccurs="unbounded">
        <xs:element name="Domicilio">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="Id" type="xs:string" minOccurs="0" />
              <xs:element name="Direccion" type="xs:string" minOccurs="0" />
              <xs:element name="Poblacion" type="xs:string" minOccurs="0" />
              <xs:element name="CodPostal" type="xs:string" minOccurs="0" />
              <xs:element name="Provincia" type="xs:string" minOccurs="0" />
              <xs:element name="Preferencia" type="xs:string" minOccurs="0" />
            </xs:sequence>
          </xs:complexType>
        </xs:element>
      </xs:choice>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

### Ejemplo:

```
<?xml version="1.0" encoding="utf-8"?>
<Domicilios>
  <Domicilio>
    <Id>87657</Id>
    <Direccion>Calle Mayor 5 4º</Direccion>
    <Poblacion>23</Poblacion>
    <CodPostal>45678</CodPostal>
    <Provincia>1</Provincia>
    <Preferencia>1</Preferencia>
  </Domicilio>
  <Domicilio>
    <Id>87658</Id>
    <Direccion>Plaza España 25 1º A</TipoVia>
    <Poblacion>23</Poblacion>
    <CodPostal>45678</CodPostal>
    <Provincia>1</Provincia>
    <Preferencia>0</Preferencia>
  </Domicilio>
</Domicilios>
```

### 3.6 Estructura de la cadena XML de lista de provincias

Métodos: **getProvinces**

DTD:

```
<!-- Lista de provincias -->
<!-- Debe existir al menos un elemento en la lista -->
<!ELEMENT Provincias (Provincia+)>

<!-- Provincia -->
<!ELEMENT Provincia (Id, Codigo, Nombre)>

<!-- Id -->
<!ELEMENT Id (#PCDATA)>

<!-- Código de provincia -->
<!ELEMENT Codigo (#PCDATA)>

<!-- Nombre de provincia-->
<!ELEMENT Nombre (#PCDATA)>
```

Esquema (XSD):

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema id="Provincias" xmlns="" xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns:msdata="urn:schemas-microsoft-com:xml-msdata">
  <xs:element name="Provincias" msdata:IsDataSet="true" msdata:Locale="es-ES">
    <xs:complexType>
      <xs:choice maxOccurs="unbounded">
        <xs:element name="Provincia">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="Id" type="xs:string" minOccurs="0" />
              <xs:element name="Codigo" type="xs:string" minOccurs="0" />
              <xs:element name="Nombre" type="xs:string" minOccurs="0" />
            </xs:sequence>
          </xs:complexType>
        </xs:element>
      </xs:choice>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

Ejemplo:

```
<?xml version="1.0" encoding="utf-8"?>
<Provincias>
  <Provincia>
    <Id>57</Id>
    <Codigo>28</Codigo>
    <Nombre>Madrid</Nombre>
  </Provincia>
  <Provincia>
    <Id>58</Id>
    <Codigo>29</Codigo>
    <Nombre>Málaga</Nombre>
  </Provincia>
</Provincias>
```

### 3.7 Estructura de la cadena XML de lista de ciudades

Métodos: **getCities**

DTD:

```
<!-- Lista de ciudades -->
<!-- Debe existir al menos un elemento en la lista -->
<!ELEMENT Ciudades (Ciudades+)>

<!-- Provincia -->
<!ELEMENT Ciudad (Id,Codigo,Nombre,IdProvincia)>

<!-- Identificador -->
<!ELEMENT Id (#PCDATA)>

<!-- Código de ciudad -->
<!ELEMENT Codigo (#PCDATA)>

<!-- Nombre de ciudad-->
<!ELEMENT Nombre (#PCDATA)>

<!-- Identificador de provincia-->
<!ELEMENT IdProvincia (#PCDATA)>
```

Esquema (XSD):

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema id="Ciudades" xmlns="" xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns:msdata="urn:schemas-microsoft-com:xml-msdata">
  <xs:element name="Ciudades" msdata:IsDataSet="true" msdata:Locale="es-ES">
    <xs:complexType>
      <xs:choice maxOccurs="unbounded">
        <xs:element name="Ciudad">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="Id" type="xs:string" minOccurs="0" />
              <xs:element name="Codigo" type="xs:string" minOccurs="0" />
              <xs:element name="Nombre" type="xs:string" minOccurs="0" />
              <xs:element name="IdProvincia" type="xs:string" minOccurs="0" />
            </xs:sequence>
          </xs:complexType>
        </xs:element>
      </xs:choice>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

Ejemplo:

```
<?xml version="1.0" encoding="utf-8"?>
<Ciudades>
  <Ciudad>
    <Id>57</Id>
    <Codigo>28017</Codigo>
    <Nombre>Madrid</Nombre>
    <IdProvincia>28</IdProvincia>
  </Ciudad>
  <Ciudad>
    <Id>58</Id>
    <Codigo>29001</Codigo>
    <Nombre>Málaga</Nombre>
    <IdProvincia>29</IdProvincia>
  </Ciudad>
</Ciudades>
```

### 3.8 Estructura de la cadena XML de lista de tipo de documentos

Métodos: **getDocsType**

DTD:

```
<!-- Lista de ciudades -->
<!-- Debe existir al menos un elemento en la lista -->
<!ELEMENT TipoDocumentos (TipoDocumento +)>

<!-- Provincia -->
<!ELEMENT TipoDocumento (Id, Descripcion, TipoPersona,Codigo)>

<!-- Identificador -->
<!ELEMENT Id (#PCDATA)>

<!-- Código de ciudad -->
<!ELEMENT Codigo (#PCDATA)>

<!-- Nombre de ciudad-->
<!ELEMENT Descripcion (#PCDATA)>

<!-- Identificador de provincia-->
<!ELEMENT TipoPersona (#PCDATA)>
```

Esquema (XSD):

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema id="TipoDocumentos" xmlns="" xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns:msdata="urn:schemas-microsoft-com:xml-msdata">
  <xs:element name="TipoDocumentos" msdata:IsDataSet="true" msdata:Locale="es-ES">
    <xs:complexType>
      <xs:choice maxOccurs="unbounded">
        <xs:element name="TipoDocumento">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="Id" type="xs:string" minOccurs="0" />
              <xs:element name="Descripcion" type="xs:string" minOccurs="0" />
              <xs:element name="TipoPersona" type="xs:string" minOccurs="0" />
              <xs:element name="Codigo" type="xs:string" minOccurs="0" />
            </xs:sequence>
          </xs:complexType>
        </xs:element>
      </xs:choice>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

Ejemplo:

```
<?xml version="1.0" encoding="utf-8"?>
<TipoDocumentos>
  <TipoDocumento>
    <Id>2</Id>
    <Descripcion>NIF</Descipncion>
    <TipoPersona>1</TipoPersona>
    <Codigo>N</Codigo>
  </TipoDocumento>
  <TipoDocumento>
    <Id>5</Id>
    <Descripcion>Otros</Descipncion>
    <TipoPersona>0</TipoPersona>
    <Codigo>X</Codigo>
  </TipoDocumento>
</TipoDocumentos>
```

### 3.9 Estructura de la cadena XML de lista de tipo de dirección

Métodos: **getAddressesType**

DTD:

```
<!-- Lista de ciudades -->
<!-- Debe existir al menos un elemento en la lista -->
<!ELEMENT TipoDirecciones (TipoDireccion +)>

<!-- Provincia -->
<!ELEMENT TipoDireccion (IdTel, Descripcion,Codigo)>

<!-- Identificador -->
<!ELEMENT Id (#PCDATA)>

<!-- Código de ciudad -->
<!ELEMENT Codigo (#PCDATA)>

<!-- Nombre de ciudad-->
<!ELEMENT Descripcion (#PCDATA)>
```

Esquema (XSD):

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema id="TipoDirecciones" xmlns="" xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns:msdata="urn:schemas-microsoft-com:xml-msdata">
  <xs:element name="TipoDirecciones" msdata:IsDataSet="true" msdata:Locale="es-ES">
    <xs:complexType>
      <xs:choice maxOccurs="unbounded">
        <xs:element name="TipoDireccion">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="Id" type="xs:string" minOccurs="0" />
              <xs:element name="Descripcion" type="xs:string" minOccurs="0" />
              <xs:element name="Codigo" type="xs:string" minOccurs="0" />
            </xs:sequence>
          </xs:complexType>
        </xs:element>
      </xs:choice>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

Ejemplo:

```
<?xml version="1.0" encoding="UTF-8"?>
<TipoDirecciones>
  <TipoDireccion>
    <IdTel>1</IdTel>
    <Descripcion><![CDATA[Teléfono (fijo)]]></Descripcion>
    <Codigo>TF</Codigo>
  </TipoDireccion>
  <TipoDireccion>
    <IdTel>2</IdTel>
    <Descripcion><![CDATA[Correo electrónico]]></Descripcion>
    <Codigo>CE</Codigo>
  </TipoDireccion>
</TipoDirecciones>
```